ISSUE 1

THE BEEBON

[illegible] – [illegible]

[illegible] – [illegible]

[illegible]

[illegible]

THE BEEBON
ISSUE NO.3 1982

AVAILABLE BY SUBSCRIPTION.
SEVEN POUND FIFTY PER YEAR.

COPYRIGHT
NOTICE
THE MAGAZINE "THE BEEBON" ITS ENTIRE CONTENTS AND NAME ARE STRICTLY COPYRIGHT OF BUG-BYTE (C) 1982. BUG-BYTE HEREBY GRANTS TO THE PURCHASER OF THIS MAGAZINE A NON-EXCLUSIVE AND NON-TRANSFERABLE LICENCE TO COPY AND USE ALL INFORMATION AND PROGRAMS CONTAINED WITHIN FOR PERSONAL USE ONLY. ALL COMMERCIAL RIGHTS ARE PRESERVED.

The BEEBON is published bi-monthly on the third week of the month. All correspondence should be addresses to:-
THE EDITOR
THE BEEBON
106, THE ALBANY
OLD HALL STREET
LIVERPOOL
MERSEYSIDE
L3 9EP

Any letters which require a reply should be accompanied by a S.A.E As we are extremly busy those enquiries which are phrased so as to make them simple to answer will solicit the fastest response.

Programs and articles sub-mitted for publication should preferably be typed or computer printout. Clear handwriting is also acceptable. All programs must be on tape or disk and have clear instructions for use and as much supporting documentation as possible. A day-time telephone number would be appreciated to allow for easier contact.

Payment is ten pounds per published page with part pages earning a proportional amount, we are prepared to negotiate for any very exceptional material.

EDITORIAL

The articles which have appeared in the first two issues of the BEEBON were written in house along with the editorial,etc. This issue its different as we have presented the best out of the contributed articles which we have recived over the last few months. In the last issue you may recall that I mentioned a BBC user group starting up on Merseyside, I am pleased to say that this group has now come into existence serving a large area of the North-West. The group meet twice a month in two different locations, the first is held at the STEM buildings in Liverpool on the third Thursday of each month, the next meeting being on October, 21st at the following address:-

STEM
117, Grove St,
By Myrtle St,
LIVERPOOL
FROM 7.30 PM

The next meeting of the month is held at Birkenhead technical college some time this month for details ring Bob Perrigo, it is usually held at this address:-

BIRKENHEAD COLLEGE OF TECHNOLIGY,
BOROUGH RD,
BIRKENHEAD

YOUR LETTERS

Each month we will print a selection of letters which we consider of interest to owners of BBC micros. Some will be constructive, we will be as unbiased as possible. Writers of all printed letters will receive a BUG-BYTE tee-shirt.

Dear Sirs,

Thank you for two excellent magazines. The games Blitz and Thargoids are extremely good, especially the sound and graphics of Thargoids and the utility programs are extremly useful.

The only programming problems I have encountered were with Thargoids. At line 600 & 630 you have opened inverted commas to print blanks to the end of the line. I know this works on most micros, but the BEEB does not like this, returning an appropiate error message. However this is easily corrected by adding a second set of inverted commas at a suitable point to blank out the previous contents of the line.

The other problem was the speed at which the setting-up routine in lines 390 to 600 ran. This happened so quickly that it was impossible to read the heading produced by line 390,470 and 550. I therefore introduced the following delays so I could watch it all happening.

```
 445 PROCSLOW
 535 PROCSLOW
 595 PROCSLOW:PROCSLOW
1395 PRINT TAB(0,0(;
1650 DEFPROCSLOW
1660 TIME=ST:REPEAT UNTIL
     TIME=ST+50
1670 ENDPROC
```

I wish the problem with printed listings from other sources were as easy to debug as this!

As I have a model B, I thought that a few simple changes should allow me to run the program in colour in Mode 1 and thus do full justice to your splendid graphic characters. The extra horizontal lines do close the display up slightly in a vertical direction but not enough to make it worthwhile altering the height of the display.

For anyone still struggling with the commands for colour on the BEEB, the following mods will give the colours of my choice. Any others can be chosen by altering the second bytes of the VDU 19 commands in line 5.

```
5 MODE 1: VDU
19,0,7,0,0,0,19,1,1,0,0,0,19,2,
11,0,0,0,19,3,0,0,0,0
 20 Delete MODE6 from this
line
 165 COLOUR 3
 385 COLOUR 2
 465 COLOUR 1
 545 COLOUR 3
 965 IF TYPE=2 THEN COLOUR 1
ELSE COLOUR 3
```

This gives black (colour 3) text on a white (colour 0) background, flashing blue and yellow gloop holes, red (colour 1) Thargoids and a black person.

An addictive game which runs fast very well - I can't keep my wife away from it!

Unlike many more unfortunate people , I had no problem to date with my BEEB apart from a 7 month delivery period! However letters to Kettering (the phone is always engaged) about the new 1.0 ROM re: its cost and how it is to be fitted and about some reimbursment for the loss of 7 months interest on my money have elicted no response. Has anyone out there got any answers to this?

Looking forward to the future do you have any plans to review printers and thier interfacing to the BEEB. I am sure this would interest a lot of people. Alas an article or two on obtaining the special effects and graphics of the BEEB in machine code routines would be appreciated.

Thanks again for producing such a good magazine.

M. Benson

ED: Thankyou for the complimentry letter and the answers to some of the problems in THARGOIDS. As to your questions on the new ROM, I am afraid we know as much as yourself but will try and keep you informed.

Dear Sirs,

I have enclosed a short graphical program for the BBC Model B which I would like you to publish in your magazine. The result is as far as I know an undiscovered product. The triangles are striped and when one is drawn on top of another they merge giving an unusual effect.

A.D.Lindsey(AGE 11)

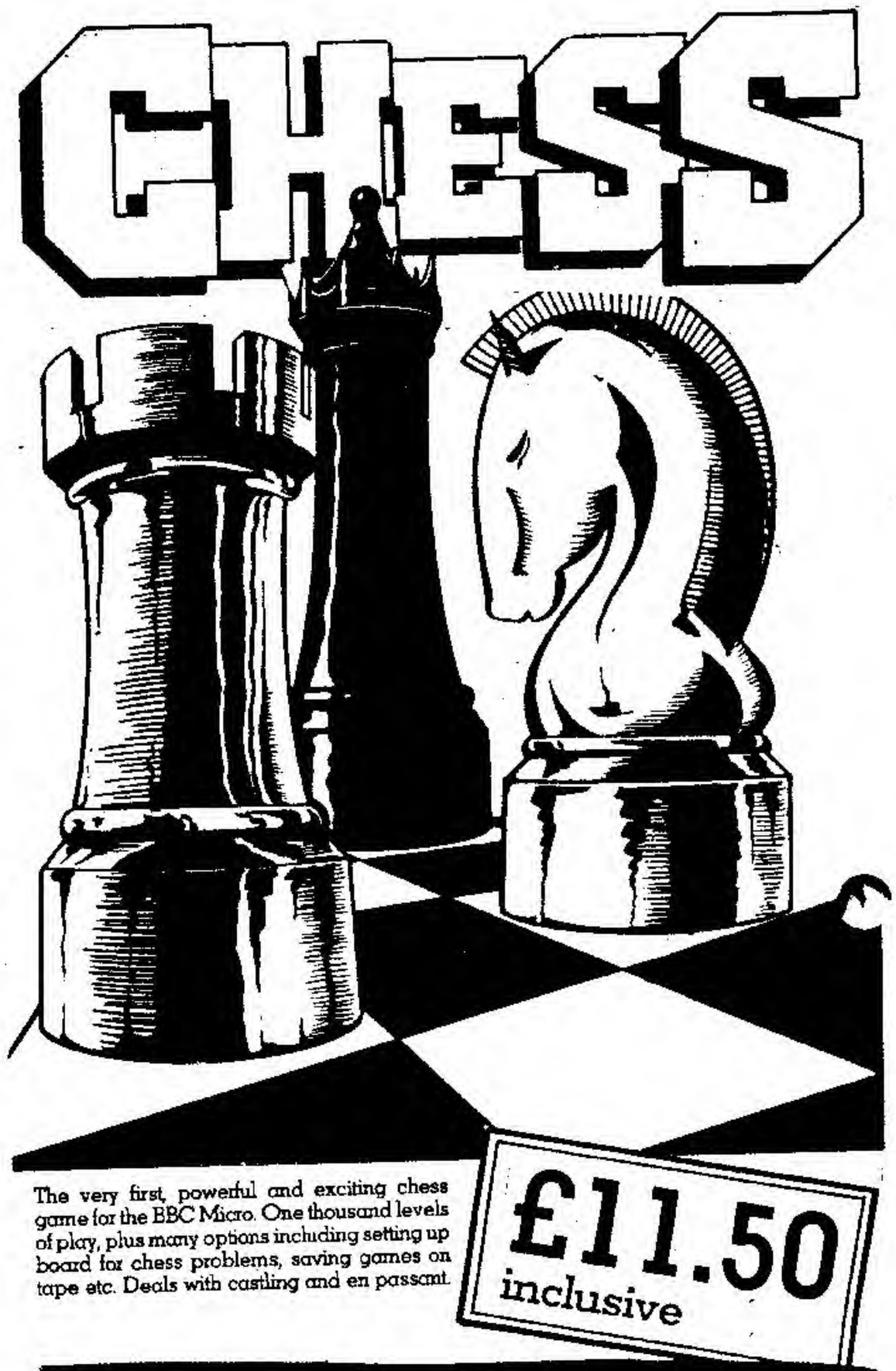
SHADES

```
10 MODE 2
20 GCOL RND(16),RND(7)
30 PLOT 85,RND(1280),RND(1024)
40 GOTO 20
```

NEW BUG-BYTE SOFTWARE NEWS

Look in the national magazines for a full page advert of our brand new adventure game "DRAGONQUEST". This full feature adventure game includes sound effects and full graphics representation of the creatures which you have to attack or the room you are currently in. The game consists of two levels, the first of which is available now, the second follows in a month or two. Both will be available for eleven pound fifty direct from us or through a local BBC dealer.

Two other new games have been introduced into our range of BBC software. The first of these is a full feature FRUIT MACHINE game, it uses the graphics and sound capibilities of the BEEB to their fullest. It's features include smooth scrolling tumblers, HOLD buttons and NUDGE facilities.



APPLE RECEIVER FROM BEEB AND APPLE TO BEEB TRANSMITTER

```
]LIST

 10  REM     NK 3 OCTOBER'82
 20  REM
 30  REM    RS232 IN SLOT2
 40  REM  8 DATA,NO PARITY,1 STOP
 50 R = 49316: POKE R,3: POKE R,21

 60  GOTO 380: REM IN LIEU OF MENU

 70  REM PRINTER IN SLOT 4
 80  CALL 768
 90  GET A$: IF A$ <  > CHR$ (27)
       THEN  PRINT A$;: GOTO 90
100  END
110  REM  BEEB TO APPLE DISK
120  CALL 768
130  DIM A$(200)
140 E$ =  CHR$ (27): REM  ESCAPE
150 R$ =  CHR$ (13): REM  C/R
160  FOR N = 1 TO 500
170 T = N / 5: IF T =  INT (T) THEN
     T =  FRE (T)
180 A$ = ""
190  GET G$
200  IF G$ = E$ THEN M = N:N = 50
     0: GOTO 230
210  IF G$ <  > R$ THEN A$ = A$ +
     G$: GOTO 190
220 A$(N) = A$:M = N
230  NEXT N
240  PRINT  CHR$ (7); CHR$ (7)
250  INPUT " FILE NAME ? ";A$
260  IF  LEN (A$) = 0 THEN 240
270 A$ = A$ + ".BEEB"
280  PRINT
290  PRINT  CHR$ (4)"OPEN"A$
300  PRINT  CHR$ (4)"WRITE"A$
310  FOR N = 1 TO M
320  PRINT A$(N)
330  NEXT N
340  PRINT  CHR$ (4)"CLOSE"A$
350  END
360  REM
370  REM  APPLE DISK TO BEEB
380  DIM A$(200)
390  ONERR  GOTO 480
400  PRINT
410  INPUT " FILE NAME ? ";A$:A$ =
     A$ + ".BEEB"
420  PRINT  CHR$ (4)"OPEN"A$
430  PRINT  CHR$ (4)"READ"A$
440  FOR N = 1 TO 200
450  GET G$: IF G$ <  > CHR$ (13
     ) THEN A$ = A$ + G$: GOTO 45
     0
460 A$(N) = A$:A$ = ""
470  NEXT N
480 M = N
490  PRINT  CHR$ (4)"CLOSE"
500  PRINT  CHR$ (4)"PR#2"
```

```
510  FOR N = 1 TO M
520  PRINT A$(N)
530  NEXT N: END
```

APPLE MACHINE CODE SUBROUTINE

```
JCALL-151

*300LL

0300-   A9 0B        LDA   #$0B
0302-   85 38        STA   $38
0304-   A9 03        LDA   #$03
0306-   85 39        STA   $39
0308-   4C EA 03     JMP   $03EA
030B-   A9 15        LDA   #$15
030D-   8D A4 C0     STA   $C0A4
0310-   AD A4 C0     LDA   $C0A4
0313-   4A           LSR
0314-   B0 0A        BCS   $0320
0316-   AD 00 C0     LDA   $C000
0319-   10 F5        BPL   $0310
031B-   8D 10 C0     STA   $C010
031E-   30 03        BMI   $0323
0320-   AD A5 C0     LDA   $C0A5
0323-   48           PHA
0324-   A9 55        LDA   #$55
0326-   8D A4 C0     STA   $C0A4
0329-   68           PLA
032A-   09 80        ORA   #$80
032C-   60           RTS
032D-   EA           NOP
032E-   EA           NOP
032F-   EA           NOP
0330-   00           BRK
0331-   00           BRK
```

BEEB TRANSMITTER AND RECEIVER

```
PROG.BEEBLIST
 1000 *FX8,4
 1010 *FX7,4
 1020 *FX5,2
 1040 *KEY9?&0210=&00:?&0211=&0D |L |M
 1050 P%=&0D00
 1060 [
 1070 .START
 1080 LDA#&02
 1090 JSR &FFEE \ open gate.
 1100 .LOOP
 1110 LDA &FE08 \ get ACIA status.
 1120 LSR A     \ check BIT 0 = 1
 1130 BCC LOOP  \ ELSE try again.
 1140 LDA &FE09 \ get ACIA data.
 1150 PHA       \  hide it
 1160 LDA #&03
 1170 JSR &FFEE \ close gate
 1180 PLA       \ recover data
 1190 RTS       \  and exit.
 1200 ]
 1210 END
```

BBC BIGPRINT
BY TONY LORD(BERKS)

If you have played with Mode 7 graphics you'll know that you can print double height characters. You can see what a dramatic impact that can have on your displays. However, you can't do that in other modes and it would be useful! The program below is designed as a PROC useable in Modes 1 and 4 directly and by changing some parameters also can be adjusted for Modes 0,2 and 5. The PROC is flexible in that it will copy any character (including user defined ones) to any Magnification you like in the colours that it is printed. It still retains the 8x8 matrix so very large characters (MAG=25 say) appear as huge blocks.

BIGPRINT

Data:	Left hand X column of new char.	(LX)
	Bottom Y row of new char	(BY)
	Character	(A$)
	Magnification multiplier	(MAG)
Action:	Check Magnification Factor > = 1	
	Print character within permitted field	PRINCHAR
	Scan character for colour information and Store in CHARSTORE Matrix	SCANCHAR
	Draw Magnified Character	MAGCHAR

PRINCHAR

Data:	Left hand X of permitted field	(LX)
	Bottom Y of permitted field	(BY)
	Character	(A$)
Constant:	X offset for LH col of char from L X	
	Y offset for bottom row of char from BX	
	(adjust for MODE)	
Action:	Link cursors	
	Move graphics cursor to right position (LX+OX, BY+OY)	
	Print character	

SCANCHAR

Data:	Left hand X of permitted field	(LX)
	Bottom Y of permitted field	(BY)
	Row increment	(RI)
	Column increment	(CI)
	(adjust RI,CI for MODE)	

Action: Define Y co-ordinate as BY+½RI
Repeat
Define X co-ordinate as LX+½CI
Repeat
Get POINT value of X,Y and store in CHARSTORE (X,Y)
Increment X by CI
Until done for all 8 columns
Increment Y by RI: Reset X co-ordinates
Until done for all 8 rows

MAGCHAR

Data:
Left hand X of permitted field (LX)
Bottom Y of permitted field (BY)
Row increment (RI)
Column increment (CI)
Magnification multiplier (MAG)

Technique: Plots quadrilaterals of colour by plotting & filling in squares of colour.
(This could be speeded up by plotting adjacent similarly coloured blocks at the same time!)

Action: Define Y1 co-ordinate as BY
Repeat
Define X1 co-ordinate as LX:MOVE to (X,Y)
Repeat
For this square:

X2 = X1+ (Column Increment x magnification)
Y2 = Y1+ (Row Increment x magnification)

then:

MOVE to X1,Y2
Fill triangle to X1,Y1
Fill triangle to X2,Y1

and:

New X1 = old X2

Until done for all 8 columns

New Y1 = old Y2 : Reset X1=LX

Until done for all 8 rows

```
10 MODE 4; DIM A$(1),CHARSTORE(8,8)
20 PRINT "BIGPRINT DEMO PROGRAM"
30 INPUT "X,Y CO-ORDS FOR BOTTOM LH CORNER", X,Y
40 INPUT "CHARACTER FOR PRINTING", A$
50 INPUT "MAGNIFICATION(>1)", MAG
60 IF LEN(A$)<>1 OR IF MAG <1 OR X<0 OR X >  1248 OR Y<0 OR
   Y>1092 THEN RUN
70 PROC BIGPRINT(X,Y,A$,MAG)
80 END

100    DEF PROCBIGPRINT(LX,BY,A$,MAG)
110    IF M<1 THEN ENDPROC
120    PROCPRINCHAR(LX,BY,A$)
130    PROCSCANCHAR(LX,BY)
140    PROCMAGCHAR (LX,BY,MAG)
150    ENDPROC

200    DEF PROCPRINCHAR(LX,BY,A$)
210    LOCAL OX,OY
220    OX=0 : OY=28
230    VDU 5   : REM LINK CURSORS
240    MOVE(LX+OX,BY+OY)
250    PRINT LEFT$(A$,1): VDU 4    : REM SEPARATE
260    ENDPROC

300    DEF PROCSCANCHAR(LX% ,BY%)
310    LOCAL RI%,CI%,ROW%,COL%,X%,Y%
320    RI%= 4: CI%=4
330    FOR ROW%= 0 TO 7
340    FOR COL%= 0 TO 7
350    X%= LX%+ CI% DIV 2 + CI%*COL%
360    Y%= LY%+ RI% DIV 2 + RI%*COL%
370    CHARSTORE(COL%,ROW%)=POINT(X%,Y%)
380    NEXT COL%: NEXT ROW%
390    ENDPROC

400    DEF PROCMAGCHAR(LX%,BY%,MAG%)
410    LOCAL CI%,RI%,ROW%,COL%,X1%,Y1%,X2%,Y2%
420    RI%=4*MAG%: CI%=4*MAG%
430    X1%=LX%: Y1%=BY%
440    FOR ROW%0 TO 7 FOR ROW%=0 TO 7
450    MOVE X1%,Y1%
460    FOR COL%0 TO 7 FOR COL%=0 TO 7
470    GCOL 0,CHARSTORE(COL%,ROW%)
480    Y2%=Y1%+RI%: X2%=X1%+CI%
490    MOVE X1%,Y2%
500    PLOT 85,X2%,Y1%: MOVE X1%,Y1%
510    PLOT 85,X2%,Y1%
520    X1%=X2%
530    NEXT COL%
540    Y1%=Y2%: X1%=LX%
550    NEXT ROW%
560    ENDPROC
```

POSSUM

With the advent of inexpensive microcomputers special schools have not been slow to realise the potential power that has suddenly been made available to them. This short program was designed for use by physically disabled pupils, many of whom do not have the power of speech. Prior to the use of a computer with its VDU they were unable to see the results of their efforts on the conventional Possum typewriter.

This initial program allows the production of text files and it is hoped that as program development takes place the "possum interface" will be incorporated in to conventional teaching programmes. The developers also wish to explore the possibilities of Possum being used by the disabled to write their own programmes. To further this aim the majority of Possum is written in BASIC in the hope that this will allow disabled programmers to modify Possum itself - eventualy giving them control over their own environment. Only when this is so will the advantages of this new technology be being exploited to the full.

This version of Possum was produced for use within the Wirral Education Authority and we would be most grateful for any comments or offers of help from local BBC owners.

Please contact Alec Wood (Seconded Teacher Computing)
Education Office Advisers Section
Cleveland Street
Birkenhead

Possum displays the character set as a matrix and the disabled user can control a cursor from two switches. These can be the original 'Puff and Suck' or any conventional push to make microswitch. The first switch moves the cursor along the matrix and the second switch moves the cursor up. Pressing the first switch again enters the character at the cursor position.

The two programmes here were developed on a BBC model B but could run on a model A with memory and user port upgrades. They enable the handicapped person to use a computer without touching the keyboard. The first program is the "Possum Input Function". This is used in a conventional program as a function instead of an INPUT statement.

eg.

(STRING VARIABLE)=FNPOSSUM(<PROMPT>)

PROMPT can be either a variable name or a literal in quotes. At the moment these strings have to be under 40 characters long because double height characters are used on the screen.

The function has the full alphabet and upper and lower case can be switched by selecting <CA>. The full set of digits and several symbols are also displayed. Characters are colour coded. Control characters being green, letters blue, and numbers and symbols red. The matrix columns and rows are numbered in magenta. <BS> is a backspace with delete, <SP> gives a space, and <CR> acts as the normal RETURN key.

SEE DIAGRAM B ON INSERT AND PAGE 13 ONWARDS

```
 10 REM**********************************
 20 REM********** ALIEN **********
 30 REM******** BY I.CROSSLEY ********
 40 REM**********************************
 50 REM
 60 MODE5:VDU19,1,9;0;19,0,6;0;23;8202;0;0;0;
 70 COLOUR1
 80 GCOL0,2
 90 PLOT69,960,0:PLOT69,400,0:PLOT85,640,400
100 PRINTTAB(5,10)"A L I E N"
110 A=INKEY(200)
120 DIMA%(20,20):ay=0:ax=0:yy=0:yx=0
130 MODE4
140 PROCINST
150 R=RND(10)
160 FORX=1TOR:XX=RND(18):FORY=1TO19:A%(XX,Y)=-1:NEXTY,X
170   R=RND(10)
180   FORX=1TOR:XX=1+RND(18):FORY=1TO19:A%(Y,XX)=-1:NEXTY,X
190     X=RND(20):IFA%(X,1)=-1THENA%(X,0)=-1 ELSE GOTO190
200     X=RND(20):IFA%(19,X)=-1THENA%(20,X)=-1 ELSE GOTO200
210     YX=RND(10)+5:YY=RND(10)+5:IFA%(YX,YY)()-1THEN210
220     AX=RND(20):AY=RND(20):IFAX=YX AND AY=YY THEN 220
230     IF A%(AX,AY)()-1 THEN220
240     MODE1
250     PRINTTAB(3,20)"AFT"TAB(4,21)"↑"TAB(4,22)"|"
260     VDU19,1,9;0;
270     COLOUR1
280     PRINTTAB(0,4)CHR$(224);:COLOUR3:PRINT" = ALIEN":COLOUR2:PRINT'CHR$(224
);:COLOUR3:PRINT" = YOU"
290     VDU28,0,31,39,28,24,0;128;1279;1023;29,0;128;
300     MOVE320,0:DRAW992,0
310     DRAW992,672:DRAW320,672:DRAW320,0
320     FORX=320TO992STEP32:PLOT69,X,672:DRAWX,0:PLOT69,320,X-320:DRAW992,X-32
0:NEXTX
330     REM  **************************
340     GCOL0,0:PLOT69,332+yx,10+yy:PLOT69,340+yx,10+yy:PLOT85,332+yx,26+yy:PL
OT85,340+yx,26+yy
350     PLOT69,332+ax,10+ay:PLOT69,340+ax,10+ay:PLOT85,332+ax,26+ay:PLOT85,340
+ax,26+ay
360     yx=YX*32:yy=YY*32
370     GCOL0,2
380     PLOT69,332+yx,10+yy:PLOT69,340+yx,10+yy:PLOT85,332+yx,26+yy:PLOT85,340
+yx,26+yy
390     ax=AX*32:ay=AY*32
400     GCOL0,1
410     PLOT69,332+ax,10+ay:PLOT69,340+ax,10+ay:PLOT85,332+ax,26+ay:PLOT85,340
+ax,26+ay
420     CLS:PRINT"You can go ";
430     A=0:P=0:S=0:F=0
440     IFA%(YX+1,YY)=-1THENPRINT"Port ";:S=1
450     IFA%(YX-1,YY)=-1THENPRINT"Starboard ";:P=1
```

```
  460      IFA%(YX,YY+1)=-1THENPRINT"Aft ";:A=1
  470      IFA%(YX,YY-1)=-1THENPRINT"Forward ";:F=1
  480      PRINT'"  Which way ?"
  490      *FX15 0
  500      A$=INKEY$(300)
  510      IFA$="W"THENYY=YY+A:IFA=0 THEN X1=YX:Y1=YY+1:PROCBLOCK
  520      IFA$="Z"THENYY=YY-F:IFF=0 THEN X1=YX:Y1=YY-1:PROCBLOCK
  530      IFA$="A"THENYX=YX-P:IFP=0 THEN X1=YX-1:Y1=YY:PROCBLOCK
  540      IFA$="S"THENYX=YX+S:IFS=0 THEN X1=YX+1:Y1=YY:PROCBLOCK
  550      IF YX=20ORYY=0THEN MODE7:PRINTTAB(0,10)"You are out of the ventilator
shafts!"'"They have been sealed off and the alien"'"blown into outer space!":GOT
0760
  560      IFAX=YX AND AY=YY THEN VDU26,12:PRINT''''''''"                You have been eat
en by the"'"                    A L I E N !!"':GOTO760
  570      A=0:P=0:S=0:F=0
  580      IF AX+1<20 AND A%(AX+1,AY)=-1THENS=1
  590      IFA%(AX-1,AY)=-1THENP=1
  600      IFA%(AX,AY+1)=-1THENA=1
  610      IF AY-1>0 AND A%(AX,AY-1)=-1THENF=1
  620      IF AX>YX THEN AX=AX-P
  630      IF AX<YX THEN AX=AX+S
  640      IF AY>YY THEN AY=AY-F
  650      IF AY<YY THEN AY=AY+A
  660      IFAX=ax/32 AND AY=ay/32 THEN GOSUB700:GOTO660
  670      IFAX=YX AND AY=YY THEN VDU26,12:PRINT''''''''"                You have been eat
en by the"'"                    A L I E N !!"':GOTO760
  680      IFAX=20 OR AY=0THENAX=ax/32:AY=ay/32:GOTO660
  690      GOTO340
  700      ON RND(4) GOTO 710,720,730,740
  710      IF P=1 THEN AX=AX-P:RETURN
  720      IF F=1 AND AY-F>0 THEN AY=AY-F:RETURN
  730      IF S=1 AND AX+S<20 THEN AX=AX+S:RETURN
  740      IF A=1 THEN AY=AY+A:RETURN
  750      RETURN
  760      PRINT''"Do you want to play again (Y/N)?":A$=GET$:IFA$="Y" OR A$="y" T
HEN CLEAR:GOTO120
  770      IFA$<>"N" AND A$<>"n"THEN760
  780      END
  790      DEF PROCINST
  800      CLS
  810      VDU5:MOVE320,1023:PRINT"A L I E N ":MOVE320,1023:PRINT"_________"
  820      VDU4:PRINT
  830      PRINT''"You are trapped in the ships ventilator"''"system. You have no
 weapons and don't"
  840      PRINT'"know the plan of the shafts."
  850      PRINT'"There are ONLY TWO exits, one forward"''"and one port."
  860      PRINT'"The controls are:-"
  870      PRINT''"       W = AFT"
  880      PRINT'"       Z = FORWARD"
  890      PRINT'"       S = PORT"
  900      PRINT'"       A = STARBOARD"
```

```
  910      PRINT''"You have got a motion detector to show"''"you where the alien
is and where you "''"are."
  920      *FX15 0
  930      PRINT'"Press 'SPACE BAR' to start.";:REPEAT:UNTILGET$=" "
  940      ENDPROC
  950      DEF PROCBLOCK
  960      X1=X1*32:Y1=Y1*32
  970      GCOL0,3:PLOT69,332+X1,10+Y1:PLOT69,340+X1,10+Y1:PLOT85,332+X1,26+Y1:PL
OT85,340+X1,26+Y1
  980      ENDPROC
```

MR. D.B. JOUBERT wrote to us saying:

"After reading an article on sound production I decided to investigate the relationship between digits entered and the actual frequency produced from the Micro.
By connecting a frequency-counter to the speaker leads I found that entering a "four" for frequency the note C2 (130HZ) was produced. By increasing digit by four the next note C2£ (13HZ) was produced, etc.."

Refer to list below for digits, notes and frequency.

LIST OF DIGITS, NOTES, FREQUENCY.

DIGIT	NOTE	FREQ.	DIGIT	NOTE	FREQ.
4	C2	130.81	100	C4	
8	C2£	138.59	104	C4£	
12	D2	146.83	108	D4	
16	D2£	155.56	112	D4£	
20	E2	164.81	116	E4	
24	F2	174.61	120	F4	
28	F2£	185.00	124	F4£	
32	G2	196.00	128	G4	
36	G2£	207.65	132	G4£	
40	A2	220.00	136	A4	
44	A2£	233.08	140	A4£	
48	B2	246.94	144	B4	
52	C3	261.63	148	C5	
56	C3£	277.18	152	C5£	
60	D3	293.66	156	D5	
64	D3£	311.13	160	D5£	
68	E3	329.63	164	E5	
72	F3	349.23	168	F5	
76	F3£	369.99	172	F5£	
80	G3	392.00	176	G5	
84	G3£	415.30	180	G5£	
88	A3	440.00	184	A5	
92	A3£	466.16	188	A5£	
96	B3	493.88	192	B5	

SORRY.SORRY.SORRY. The £ signs shown above should really be# but the Osborne prints them out as £ signs.

```
 1 REM PROGRAM DESIGN ALEC WOOD
 4 REM*****************************************
 5 REM******POSSUM INPUT FUNCTION******
 6 REM******       I. CROSSLEY       ******
 7 REM*****************************************
 8 REM*FIRST LINE OF PROGRAM*
10 MODE7:PROCINIT
30 REM ***************************************
50 REM*Instead of INPUT"HELLO"A$ use
60 REM  A$=FNPOSSUM("HELLO")**
70 REM
80 REM**********  YOUR PROGRAM HERE ******
90 REM
100 REM*LAST PART OF PROGRAM*
110 DEF FNPOSSUM(PR$)
120 LOCALXX,YY,X,Y,P,L,INP$,A$,M,F,UC
130 RESTURE
140 ?&FE62=0
150 FORY=1TO7:FORX=1TO8
160      READA$:IFLEN(A$)<3THENA$=A$+"   "
170      IFA$="   "THENA$=CHR$(&22)+"   "
180      IFA$="   "THENA$="  "+CHR$(255)+"  "
190      CH$(X,Y)=A$:NEXTX,Y
200    FORX=0TO8:CH$(0,8-X)=STR$(X)+"   ":CH$(X,8)=STR$(X)+"   ":NEXTX
210    DATAK,Q,6,0,-,+,$,"    "
220    DATAW,V,J,5,9,=,#,"    "
230    DATAU,M,B,2,4,7,),"    "
240    DATAR,D,Y,",",X,*,"    "
250    DATA"CA.",N,C,F,".",1,(,8
260    DATAT,A,S,H,P,G,Z,""""
270    DATA"SP.",E,O,I,L,"CR.","BS.",3
280    UC=0:F=0:INP$=""
290    PROCCLS
300    PRINTTAB(0,11)CHR$(141)PR$TAB(0,12)CHR$(141)PR$
310    P=0:L=15
320    FORY=1TO8:FORX=0TO8
330        PROCPRINT(CH$(X,Y),X*4+2,Y,1)
340        NEXTX,Y
350      XX=0:YY=8:PROCPRINT(CH$(XX,YY),XX*4+2,YY,2)
360      M=FNGET
370      IFM=1 AND F=0 THENPROCPRINT(CH$(XX,YY),XX*4+2,YY,1):XX=XX+1:IFXX)HE
XX=1
380      IFM=2 AND XX()0 THENF=1:PROCPRINT(CH$(XX,YY),XX*4+2,YY,1):YY=YY-1:IFY
(1THENYY=8
390      IFM=1 AND F=1 AND YY(8 THENPROCPRINT(CH$(XX,YY),XX*4+2,YY,1):F=0:GOTO
30
400      IFM=1 AND F=1 AND YY=8 THENPROCPRINT(CH$(XX,YY),XX*4+2,YY,1):F=0:GOTO
50
410      PROCPRINT(CH$(XX,YY),XX*4+2,YY,2)
420      GOTO360
430      DEF PROCCLS
440      CLS
```

```
450       LOCALX
460       FORX=0TO9:PRINTTAB(0,X);CHR$(&83);CHR$(157);
470         NEXTX:ENDPROC
480       DEF PROCPRINT(A$,X,Y,C)
490       LOCALCOL,B$
500       IFC=1ANDLEFT$(A$,1)="A"ANDLEFT$(A$,1)<="Z"THENCOL=132 :GOTO550
510       IFC=2THENCOL=135
520       IFC=1AND(LEFT$(A$,1)<"A"ORLEFT$(A$,1)="-")THENCOL=&81
530       IF(X=20RY=8)ANDC=1THENCOL=&85
540       IFC=1ANDRIGHT$(A$,1)=". "THENCOL=&82
550       IFRIGHT$(A$,1)=" "THENB$=FNLCASE(A$)ELSEB$=A$
560       PRINTTAB(X,Y);CHR$(COL);B$;
570       PRINTTAB(P,L);
580       VDU255
590       ENDPROC
600       DEF FNLCASE(A$)
610       LOCALB$
620       IFUC=1THENB$=A$:GOTO640
630       IFLEFT$(A$,1)="A" AND LEFT$(A$,1)<="Z" THENB$=CHR$(ASC(LEFT$(A$,1))+&
20)+"   "ELSEB$=A$
640       =B$
65        DEF FNGET
660       LOCALA,B
670       NTIME=TIME:REPEAT:B=?&FE60:UNTILB=255 OR (TIME=NTIME+200)
680       A=?&FE60
690       IFA=255THEN680
700       IFA=254THENA=1
710       IFA=253THENA=2
720       =A
730       A$=CH$(XX,YY)
740       IFA$=" "+CHR$(255)+" "THEN350
750       IFRIGHT$(A$,1)=" "THENPROCCHAR(A$):GOTO350
760       IFA$="CA."THENUC=1-UC:A=USR(PROG%):GOTO350
770       IFA$="SP."THENPROCCHAR(" "):GOTO350
780       IFA$="CR."THEN =INP$
790       IFA$="BS."THENPROCBS:GOTO350
800       STOP
810       DEF PROCCHAR(A$)
820       IFUC=0THENA$=FNLCASE(A$)
830       PRINTTAB(P,L);LEFT$(A$,1);:P=POS:L=VPOS
84        INP$=INP$+LEFT$(A$,1)
850       ENDPROC
860       DEF PROCBS
870       IFINP$=""THENENDPROC
880       INP$=LEFT$(INP$,LEN(INP$)-1):PRINTTAB(P,L);" ":VDU127:VDU127:P=P-1
890       ENDPROC
900       FORY=1TO8:FORX=0TO8
910           IF(LEFT$(CH$(X,Y),1)="A"ANDLEFT$(CH$(X,Y),1)<="Z")ANDRIGHT$(CH$(X
,Y),1)="  "THENA$=FNLCASE(CH$(X,Y)):PRINTTAB(X*4+2,Y)CHR$(132)A$
920           NEXTX,Y
930         GOTO350
```

```
1000        DEF PROCINIT
1010        DIMCH$(8,8),PROG% 60
1020        VDU23;8202;0;0;0;
1030        FORC=0TO2STEP2
1040          P%=PROG%
1050          [OPT C
1060          LDX #0
1070          .LOOP1 LDA &7C28,X
1080          CMP #&84
1090          BEQ YEH
1100          .RINK1 INX
1110          CPX #&90
1120          BNE LOOP1
1130          LDX #0
1140          .LOOP2 LDA &7CB8,X
1150          CMP #&84
1160          BEQ YEH1
1170          .RINK2 INX
1180          CPX #&90
1190          BNE LOOP2
1200          RTS
1210          .YEH LDA &7C29,X
1220          CLC
1230          EOR #&20
1240          STA &7C29,X
1250          JMP RINK1
1260          .YEH1 LDA &7CB9,X
1270          CLC
1280          EOR #&20
1290          STA &7CB9,X
1300          JMP RINK2
1310          ]
1320          NEXTC
1330        ENDPROC
```

```
 5 REM PROGRAM DESIGN ALEC WOOD
 10 REM*********************************
 20 REM****     I. CROSSLEY     ****
 30 REM***POSSUM WORD PROCESSOR***
 40 REM*********************************
 50 REM    INITIALISE
 60 MODE7
 70 PROCINIT
 80 UC=1:F=0
 90 FORY=1TO7:FORX=1TO8
100       READA$:IFLEN(A$)<3THENA$=A$+"   "
110       IFA$="'   "THENA$=CHR$(&22)+"   "
120       CH$(X,Y)=A$:NEXTX,Y
130    FORX=0TO8:CH$(0,8-X)=STR$(X)+"   ":CH$(X,8)=STR$(X)+"   ":NEXTX
140    DATAK,Q,E,0,-,+,$,"TI."
150    DATAW,V,J,5,9,=,#,"TO."
160    DATAU,M,B,2,4,7,>,8
170    DATAR,D,Y,",",X,÷,*,"'"
180    DATA"CA.",N,C,F,".",1,(,3
190    DATAT,A,S,H,P,G,Z,"CL."
200    DATA"SP.",E,O,I,L,"CR.","BS.","PR."
210    PROCCLS
220    P=0:L=10
230    FORY=1TO8:FORX=0TO8
240       PROCPRINT(CH$(X,Y),X*4+2,Y,1)
250       NEXTX,Y
260    REM          ENTER AND START
270    XX=0:YY=8:PROCPRINT(CH$(XX,YY),XX*4+2,YY,2)
280    M=FNGET
290    SOUND1,-15,100*M+50,2:SOUND2,-15,100*M,2
300    IFM=1 AND F=0 THENPROCPRINT(CH$(XX,YY),XX*4+2,YY,1):XX=XX+1:IFXX>8THEN
X=1
310    IFM=2 AND XX<>0 THENF=1:PROCPRINT(CH$(XX,YY),XX*4+2,YY,1):YY=YY-1:IFYY
<1THENYY=8
320    IFM=1 AND F=1 AND YY<8 THENPROCPRINT(CH$(XX,YY),XX*4+2,YY,1):F=0:GOTO6
0
330    IFM=1 AND F=1 AND YY=8 THENPROCPRINT(CH$(XX,YY),XX*4+2,YY,1):F=0:GOTO2
0
340    PROCPRINT(CH$(XX,YY),XX*4+2,YY,2)
350    GOTO280
360    END
370    DEF PROCCLS
380    CLS
390    LOCALX
400    FORX=0TO9:PRINTTAB(0,X);CHR$(&83);CHR$(157);
410       NEXTX:ENDPROC
420    DEF PROCPRINT(A$,X,Y,C)
430    LOCALCOL,B$
440    IFC=1THENCOL=132
450    IFC=2THENCOL=135
460    IFC=1AND(LEFT$(A$,1)<"A"ORLEFT$(A$,1)="^")THENCOL=&81
470    IF(X=20RY=8)ANDC=1THENCOL=&85
```

```
480       IFC=1ANDRIGHT$(A$,1)=". "THENCOL=&82
490       IF UC=0 AND RIGHT$(A$,1)=" "THENB$=FNLCASE(A$)ELSEB$=A$
500       PRINTTAB(X,Y);CHR$(COL);B$;
510       PRINTTAB(P,L);
520       ENDPROC
530       DEF FNLCASE(A$)
540       IFUC=1THEN=A$
550       LOCALB$
560       IFLEFT$(A$,1)="A" AND LEFT$(A$,1)<="Z" THENB$=CHR$(ASC(LEFT$(A$,1))+&
20)+"   "ELSEB$=A$
570       =B$
580       DEF FNGET
590       LOCALA,B
600       NTIME=TIME:REPEAT:B=?&FE60:UNTILB=255 OR (TIME=NTIME+200)
610       A=?&FE60
620       IFA=255THEN610
630       IFA=254THENA=1
640       IFA=253THENA=2
650       =A
660       A$=CH$(XX,YY)
670       IFRIGHT$(A$,1)=" "THENPROCCHAR(A$):GOTO270
680       IFA$="CA."THENUC=1-UC:ABC=USR(PROG%):GOTO270
690       IFA$="SP."THENPROCCHAR(" "):GOTO270
700       IFA$="PR."THENPROCPR:GOTO210
710       IFA$="CR."THENPROCCR:GOTO270
720       IFA$="CL."THENIFSTORE$(LI)=""THENPROCDISP:GOTO210
730       IFA$="CL."THENPROCCR:PROCDISP:GOTO210
740       IFA$="TO."THEN 930
750       IFA$="BS."THENPROCBS:GOTO270
760       IFA$="TI."THEN1150
770       STOP
780       DEF PROCCHAR(A$)
790       IFVPOS>29THENENDPROC
800       IFUC=0THENA$=FNLCASE(A$)
810       IFA$=CHR$(13)THENPRINT:L=VPOS:P=POS
820       PRINTTAB(P,L);LEFT$(A$,1);:P=POS:L=VPOS
830       STORE$(LI)=STORE$(LI)+LEFT$(A$,1)
840       ENDPROC
850        DEF  PROCPR
860       VDU2
870       PROCDISP
880       VDU3
890       ENDPROC
900       DEF PROCCR
910       PROCCHAR(CHR$(13)):LI=LI+1
920       ENDPROC
930       IFLEFT$(STORE$(LI),1)<>"#"THEN270
940       OUT$=RIGHT$(STORE$(LI),LEN(STORE$(LI))-1)
950       PRINT'';"PRESS ";
960       CHAN=OPENOUT(OUT$)
970       FORX=1TOLI-1
```

```
 980         FORY=1TOLEN(STORE$(X))-1
 990           BPUT#CHAN,ASC(MID$(STORE$(X),Y,1))
1000           FORUC=1TO10:NEXTUC
1010           NEXTY:BPUT#CHAN,13
1020         NEXTX
1030       BPUT#CHAN,0
1040       CLOSE#CHAN
1050       CLEAR:GOTO60
1060       DEF PROCBS
1070       IFSTORE$(LI)<>CHR$(13)ANDSTORE$(LI)<>""THENPRINTCHR$(127);:STORE$(LI)=
LEFT$(STORE$(LI),LEN(STORE$(LI))-1):P=POS:L=VPOS:ENDPROC
1080       IFL>10THENSTORE$(LI)="":PROCDEC:P=LEN(STORE$(LI))-1:STORE$(LI)=LEFT$(S
TORE$(LI),P)
1090       IFP>40THENP=P-40
1100       PRINTTAB(P,L);
1110       ENDPROC
1120       DEF PROCDEC
1130       IFSTORE$(LI)=""THENLI=LI-1:L=L-1
1140       ENDPROC
1150       IFLEFT$(STORE$(LI),1)<>"#"THEN270
1160       IN$=RIGHT$(STORE$(LI),LEN(STORE$(LI))-1)
1170       FORX=1TO100:STORE$(X)="":NEXTX
1180       CLS
1190       PRINT'''";"      PRESS PLAY ON TAPE"
1200       CHAN=OPENIN(IN$)
1210       LI=1
1220       REPEAT
1230         X=BGET#CHAN
1240         IFX>0THEN STORE$(LI)=STORE$(LI)+CHR$(X)
1250         PRINTCHR$(X) ;
1260         IFX=13THENLI=LI+1:PRINT
1270         UNTIL X=0
1280       RESTORE:GOTO80
1290       DEF PROCDISP
1300       LOCALX,Y
1310       CLS
1320       FORX=1TOLI:NTIME=TIME+600:REPEATUNTIL?&FE60=253ORTIME=NTIME
1330         PRINTSTORE$(X):REPEATUNTIL?&FE60=255:NEXTX
1340       ENDPROC
1350       DEF PROCINIT
1360       *FX5,1
1370       LI=1
1380       ?&FE62=0:DIMCH$(8,8),STORE$(100),PROG% 60
1390       FORC=0TO2STEP2
1400         P%=PROG%
1410         [OPT C
1420         LDX #0
1430         .LOOP1 LDA &7C28,X
1440         CMP #&84
1450         BEQ YEH
1460         .RINK1 INX
```

```
1470        CPX #&90
1480        BNE LOOP1
1490        LDX #0
1500        .LOOP2 LDA &7CB8,X
1510        CMP #&84
1520        BEQ YEH1
1530        .RINK2 INX
1540        CPX #&90
1550        BNE LOOP2
1560        RTS
1570        .YEH LDA &7C29,X
1580        CLC
1590        EOR #&20
1600        STA &7C29,X
1610        JMP RINK1
1620        .YEH1 LDA &7CB9,X
1630        CLC
1640        EOR #&20
1650        STA &7CB9,X
1660        JMP RINK2
1670        ]
1680        NEXT C
1690    ENDPROC
```

BEEB TO TELETYPE SC/MP INTERFACE

COPYRIGHT BOB PERRIGO

ADDRESS

```
0000    08 C4 01 01 19 C4 08 35 C4 20 31 C4 08 3C
000E    C4 80 32 C4 20 C9 04 C4 00 C9 02 C4 40 C9
001C    03 C4 48 CA 00 C1 01 D4 40 98 FA C1 00 CA
002A    01 01 C4 00 01 19 01 C4 91 8F 11 C4 0A CA
0038    02 19 40 DC B0 01 C4 B6 8F 11 BA 02 9C F3
0046    BA 00 C2 01 E4 0D 9C 08 C4 49 CA 00 C4 0A
0054    90 D3 C2 00 9C C7 C4 0D 90 CB
```

After obtaining a model B the next thing anybody wants is a printer. Having looked at the bank balance after obtaining the BEEB it is usually a job for the future.

There are a lot of RS232 and 20m/a teletypes around at very low prices, which will give good but slow hard copy. They are normally 110 BAUD which is not available from the RS432 socket, and they are not intelligent. i.e. They do not count the characters per line and CR/LF on an overlapping line, or give a LF after a CR.

The interface shown gives these facilities on a teletype. The printer socket and ribbon cable came from Technomatic Ltd, the SC/MP and the INS 8154 were on the old MK14 along with a spare gate of IC 16 (74LS08) which can be used as a buffer if required, i.e. if R162 inside the BEEB is below the 4.7K as shown in the diagram, pin 1 on the INS 8154 may not drive it.

The 94 bytes of program will work in RAM or ROM. In the prototype it is in ROM so the old MK14 is working as a dedicated machine.

SEE INSERT SIDE A.

I N T E R L I N K

by Nik KELLY) @ 4 October 1982

This article describes a 1200 Baud link between a BBC'B' and an Apple.

Interlink is very simple. If I want to send information from Apple to Beeb, a short program makes the Beeb behave like a printer. To reverse the process, I reset both computers and run a similar program into the Apple.

HARDWARE

The hardware required is not complicated as both the Beeb 'B' and the serial card in my Apple use the 6850 ACIA. This ingenious chip handles the coding and decoding, parity, stop bits and all, through two memory locations for ' Read Status -- Write Control ' and ' Read In -- Write Out '. The serial card came with a ribbon cable to a D25 socket and I added a short cable with 'domino' and D25 plugs. I took care to swop the CTS/RTS and DIN / DOUT lines. I also found that I had to put a mark on the Beeb and its plug, as the the 'domino' plug was reversible.

SOFTWARE

I'll describe the Apple's software first. Both computers use the 6502 processor chip, so the machine code is very similar. Unfortunately, Acorn and Apple use slightly different mnemonics for some instructions, LSR for LSR A , for example.

Calling the short machine code program at 768, HEX 300, connects the second part, commencing HEX 30B , as the Apple's source of input. For every character received, this main program puts HEX 15 into the ACIA Status register to open the data gate. It then enters a loop.

First it reads the ACIA status. If a character has been received, bit zero is set to one. This is easily checked by shifting it into the Carry.

The program would branch on 'Carry Set' to read the data register, then close the gate. If the ACIA is empty, the program checks the Apple's keyboard, which also uses two addresses. When a key is pressed, bit 7 is set to one in HEX C000 , making it negative in signed binary, while the lower bits carry the ASCII code. If that location is still positive, the program branches to the start of the loop. Otherwise, it resets HEX C000 by accessing HEX C010 then skips down to rejoin the serial route. In both cases, the character is temporarily stored on the stack while the ACIA gate is closed.

As I've bypassed the normal Apple input route, which would have converted all lower-case letters to upper-case, I have to check incoming data in my program. The Apple stores BASIC commands in coded form, one letter per key word, but distinguishes them from strings etc by setting the bit 7 to one. To prevent confusion, the program masks this bit to zero before passing the data to the Apple. You must re-connect the Apple's normal operation with PR # 0 after the transfer is complete, or strange things happen when you try to use its edit functions !

The associated BASIC program uses the machine code to GET one character at a time from the serial interface, and build up a string. This is necessary because the Apple does not have an INPUT LINE command, and INPUT would be confused by commas, colons etc. An ASCII 13 is the end-of-line marker, and that complete string is then stored in an array. An 'Escape' key, ASCII 27 from the Apple's keyboard or VDU 1, 27 from the Beeb signals the end-of-file. The Apple then chimes twice and prompts for a file name, writing the string array to disk as a text-file.

Reading it back is a very similar process. The program again uses GET to read the disk file, stores the lot in an array, then sends it to the 'printer' that it thinks is connected to the serial card.

At the Beeb, though, things are a bit more complicated because there is an inscrutable UIA between you and the ACIA. I must thank Acorn for explaining how to make the RS423 show 'Busy' and close the gate, but I still don't understand why the design doesn't permit easier control. I suspect that the 100 interrupts per second from the TIME clock may be responsible, in which case there should be a 'Serial Register Back-up Byte' somewhere. Changing the contents of that would have the same

effect as the direct access in the Apple. It would certainly make the Beeb's serial input a lot faster !

The present BASIC program sets up the baud rate, and assembles the machine code in a convenient place. I've followed the example in Acorn User, with the JIM page for the code, and a defined key to connect the link, but a DIM P% & 16 on line one would do if the incoming program was higher numbered, or data was flowing through an INPUT statement.

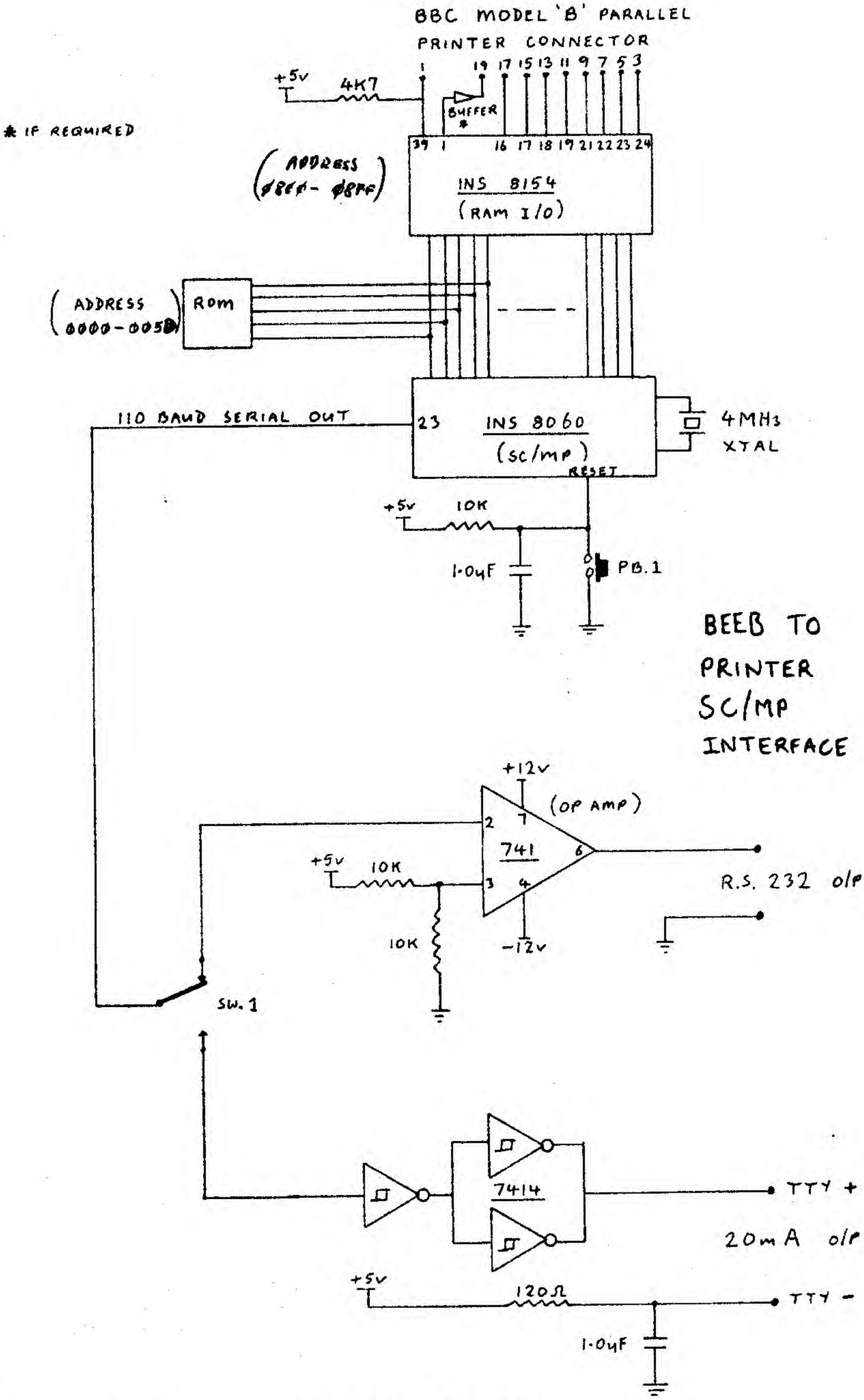
The program opens the ACIA by sending a Control-B (Printer On). It then loops until a character is caught, when it stores the data on the stack and closes the ACIA gate by sending a Control-C (Printer Off).

To transfer from Apple to Beeb, you must run the set-up program, press f9 to put the Beeb into its LISTEN mode, run the Apple program to send the data across, then recover control with the BREAK key and OLD. Please check that the first line of the program is intact, as the first character received by the Beeb may be scrambled. If much data is lost, check the serial card parameters in the other computer. The ACIA should be set to 8 bits data, no parity and one stop bit, and the serial card should respond to the Beeb's CTS/RTS signals. Please consult manuals, friends etc in case of difficulty. If all else fails, check the wiring. I had mine back-to-front the first time !

Interlink was demonstrated at the September meeting of the Liverpool BBC & Atom group, and represents the first stage in a simple serial network. If you have modifications, improvements or suggestions, please send them to Beebon, or bring them along to us. We meet at STEM, 117 Grove St, by Myrtle St, Liverpool between 7.30 and 9.30 PM on the Third Thursday of each month. Comments from Beeb users with OS 1.0 would be particularly welcome !

BUG-BYTE SOFTWARE

(A)



© BOB PERRIGO DIAGRAM LES PICKSTOCK

The second program is the "Possum Word Processor". This is a self contained program and has printer and tape output routines. As yet there are no editing facilities other than the (BS) delete. The tape facility enables a program written using Possum to be loaded back into the computer using *EXEC. At the moment this has to be done via the normal keyboard by an able person. The format for saving to tape and for loading is to enter a file name preceded by a hash # .

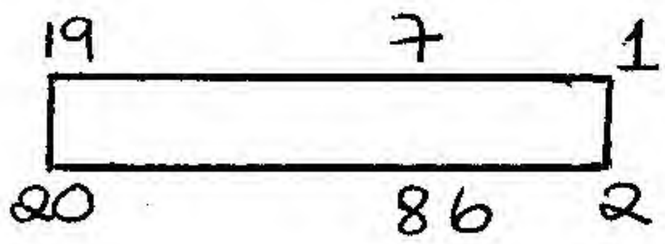
eg.

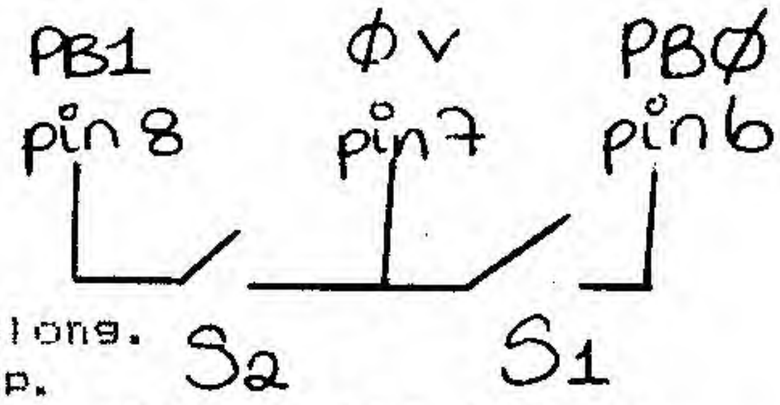
#FILE1

This filename must be on a clear line. Tape Input or Output is then achieved by selecting (TI) or (TO). (TI) will also load *SPOOLed data.

When the screen becomes filled with text it must be cleared using (CL). This displays ALL text so far entered and then clears the screen still remembering the text. (PR) does the same thing but sends the output to a printer connected to the parallel port as well.

The diagram below shows how the microswitches should be connected to the user port. Pins 6, 7 and 8 are respectively used as switch1, Earth, and switch2.





SUMMARY

S1 moves the cursor along.
S2 moves the cursor up.
S1 then enters the character at the cursor.
(CR) = RETURN
(SP) = SPACE
(CA) = CHANGE ALPHA
(BS) = BACK SPACE
(CL) = CLEAR SCREEN
(PR) = OUTPUT TO PRINTER
(TO) = OUTPUT TO TAPE
(TI) = INPUT FROM TAPE

(CL) and (PR) wait after each screen line for S2 to be pressed. This is to allow the user time to read each line if required.

These two programs could not have been written without information given by the staff of Clatterbridge Special School, where field trials are now taking place. In the future we are hoping to make the listings available in space provided on PRESTEL for educational users.

I. CROSSLEY.